Super-Resolution Imaging

List of Figures

1.1 Super-resolution can be applied to video sequences by combining the frames in (overlapping) groups of N consecutive frames. 3

1.2 Comparison of frequency responses (PSF and sensor integration) for different imaging systems. Simulated frequency responses are shown for camera systems with a 10.1 Mpixel $4/3''$ sensor and $f/5.6$ and $f/2.8$ aperture of the optics, a 3.2 Mpixel $1/2.5''$ sensor with lens aperture $f/2.8$, and a QCIF video obtained by combining blocks of pixels from the 3.2 Mpixel $1/2.5''$ sensor. 5

1.3 Measured frequency responses for a Leica DC250 camera with a Nikon 85mm optical system and a Sigma SD10 camera with a Sigma 18-50 mm lens. 6

1.4 Resolution chart used to measure image resolution according to the ISO standard [10]. 7

1.5 Three sampling situations can be distinguished: (a) Nyquist sampling ($K > 2M$), (b) Partially aliased signals ($M < K < 2M$), and (c) Totally aliased signals ($K < M$). 11

1.6 Example of the decomposition of the objective function into its different components belonging to orthogonal subspaces. . . . 12

1.7 Simulation results of the algorithms for totally aliased images (noiseless). (a) One of the five 16×16 images used as input. (b) Reconstructed 31×31 image using the algorithm from Section 1.4.1. (c) One of the five 32×32 images used as input. (d) Reconstructed 63×63 image using the algorithm from Section 1.4.2. 14

1.8 Examples of the objective functions in (1.15) and (1.19). (a) Two sets of 91 samples, with 81 unknown coefficients used in (1.15). The exact offset is $t_1 = 54.6$. Next to the global minimum, they also contain many local minima. (b) Three sets of 41 samples, with 81 unknown coefficients used in (1.15). The exact offsets are $t_1 = 8.2$ and $t_2 = 24.6$. Small values are represented by dark pixels. (c) Two sets of 91 samples, with 81 unknown coefficients used in (1.19). The exact offset is $t_1 = 54.6$. (d) Three sets of 41 samples, with 81 unknown coefficients used in (1.19). The exact offsets are $t_1 = 8.2$ and $t_2 = 24.6$. Small values are represented by dark pixels. 15

1.9 The amplitude of the Fourier transform of an image is rotated over the same angle ($\theta_1 = 25°$) as the spatial domain image. (a) Original image. (b) Rotated image. (c) Fourier transform amplitude of the original image. (d) Fourier transform amplitude of the rotated image. 18
1.10 Results using the frequency domain registration algorithm for partially aliased images on a set of still images taken with a Leica DC250 camera. (a) Part of one of the input images, with a detail showing the aliasing in (b) and (c). (d) High resolution image reconstructed from part of four Leica DC250 input images. Details are shown in (e) and (f) to display the differences better. 19
1.11 Results of the frequency domain registration algorithm for partially aliased images on the Sigma SD10 images of the outdoor scene. (a) (Part of) one of the 4 input images, with a detail in (b). (c) High resolution output image, with a detail of the central part in (d) to show the differences better. 20
1.12 Example of signal registration for a 1D case. (a) A signal composed of a sinusoid and a step function is uniformly sampled twice with an unknown shift δ (in this example $\delta = 0.083$). The shift between the two sets of samples should be determined to reconstruct the original signal. (b) and (c) Low-pass interpolation of the two sets of samples. (d) Difference between the signals from (b) and (c) for the correct shift ($\delta = 0.083$). The error is large around the discontinuity (outlier) and small elsewhere. (e) Difference between the signals in (b) and (c) for the shift obtained using MSE minimization ($\delta = 0.25$). The error is minimized over the entire domain, but an incorrect shift value is obtained. 27
1.13 Images used to test the motion estimation algorithm. (a) Original high resolution (3072×2304) image. (b) Low resolution reference image (176×144). (c) One of the 100 low-resolution non reference images (176×144) registered with the reference image in (a). 28
1.14 Results of super-resolution on low-rate encoded videos. The first column shows one video frame, the second column shows the result of bicubic interpolation, and the third column the result of the super-resolution algorithm applied to 9 consecutive video frames. 29

1

Registration for Super-Resolution: Theory, Algorithms, and Applications in Image and Mobile Video Enhancement

Patrick Vandewalle

Philips Research

Luciano Sbaiz

Google

Martin Vetterli

Ecole Polytechnique Fédérale de Lausanne

CONTENTS

1.1 Camera model 3
1.2 What is Resolution? 6
1.3 Super-Resolution as a Multichannel Sampling Problem 7
1.3.1 Fourier series 8
1.4 Registration of Totally Aliased Signals 10
1.4.1 Variable Projection Method 10
1.4.2 Frequency Analysis Method 12
1.4.3 Results 13
1.5 Registration of Partially Aliased Signals 13
1.5.1 Super-resolution using frequency domain registration 16
1.5.1.1 Image registration 16
1.5.1.2 Image reconstruction 17
1.5.1.3 Results 17
1.5.2 Super-resolution from low quality videos 21
1.5.2.1 Motion model 21
1.5.2.2 Image registration 22
1.5.2.3 Image reconstruction 23
1.5.2.4 Results on video sequences 25
1.6 Conclusions 26

The relation between sensor resolution and the optics of a digital camera is determined by the Nyquist sampling theorem: the sampling frequency should be larger than twice the maximum frequency of the image content coming out of the optical system. If a lower resolution is used, the output is aliased. Aliasing in digital images is often considered as a nuisance and (both optical and digital) filters are designed to avoid aliasing in digital cameras. However,

aliasing also contains extra high-frequency information with additional details about the scene. Super-resolution algorithms extract the information present in the aliasing to reconstruct a higher resolution image.

Super-resolution algorithms typically combine multiple aliased images with small relative motion, and create a single high resolution image. The input can be a set of pictures taken with a digital camera from approximately the same point of view. An application could be to use a low resolution camera (with a good optical system), capture a set of images while holding the camera manually in approximately the same position, and use the small camera shake to reconstruct a high resolution image. This would allow to take multiple images with a cheap camera, and combine them to a higher resolution image as if it had been taken with a more expensive camera. Other applications can be found for example in situations where a camera sensor can not be easily replaced, such as in satellites. It is (almost) impossible to install a new camera sensor, while a modification of the software allows to take a series of images of approximately the same subject.

The set of images used in a super-resolution algorithm can also be (part of) a video sequence, where the motion between subsequent frames is typically small. An application can be found in upscaling of low resolution videos, such as those acquired with handheld devices. These devices are typically able to acquire videos with low resolution (such as CIF, i.e. 288×352 pixels, or QCIF, i.e. 144×176 pixels) and the videos are coded at relatively low bit rates (such as 128 Kb/s). Despite the low quality of these videos, we will show later that super-resolution algorithms can be applied, under certain hypotheses, to increase the resolution and obtain additionally a significant reduction of coding artifacts. The procedure to apply a super-resolution algorithm to a video sequence is represented in Figure 1.1. The input frames are combined in groups of N consecutive frames. One of the frames, for example I_0, is estimated at higher resolution to produce the output frame I'_0. The result is added to the output sequence and the procedure is repeated taking the next input frame as a reference. A simple way to manage the input frames is to use a circular buffer containing the N most recent input frames. A more flexible approach, not considered here, is to use a buffer of variable size. This would allow to process sequences with varying speed and scene changes.

Super-resolution has been a very active research topic over the past few decades. In 1984, Tsai and Huang introduced a first super-resolution algorithm to reconstruct a high-resolution image from multiple shifted low-resolution images using a frequency domain approach [16]. A good overview of existing super-resolution algorithms is given by Borman and Stevenson [2] and Park et al. [13], or in special issues on the topic in IEEE Signal Processing Magazine (edited by Kang and Chaudhuri [9]) and EURASIP Journal of Applied Signal Processing (edited by Ng et al. [3]).

Most super-resolution algorithms consist of two main parts: image registration, where the images are precisely aligned, and image reconstruction, where the aligned images are combined to estimate a higher resolution image.

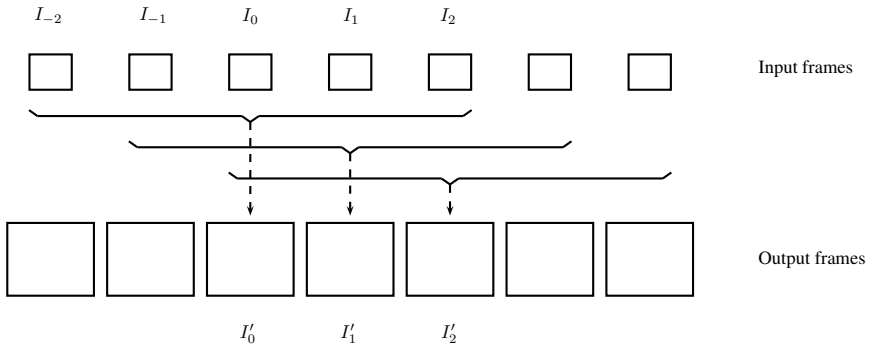


FIGURE 1.1
Super-resolution can be applied to video sequences by combining the frames in (overlapping) groups of N consecutive frames.

In this chapter, we will concentrate on the first part, as precise, sub-pixel image registration is needed in order to be able to correctly reconstruct any high resolution information. For the reconstruction, we will use existing approaches.

In the next section, we first discuss our camera model, and how super-resolution can be applied to images captured with such cameras, followed by a definition of what we understand by the term 'resolution'. We will then present super-resolution as a multichannel sampling problem with unknown offsets. Using this description, an analysis can be made about the nature of the problem and conditions under which a solution can be found. Next, we describe a few solution methods using subspace approaches. We describe two solution methods for registration of totally aliased signals, followed by two more efficient methods that take advantage of aliasing-free parts of the input images to perform sub-pixel registration. The work presented in Sections 1.1-1.5.1 was already presented earlier [17, 18, 19] and is reproducible. The code and data to reproduce those results can be downloaded from the cited websites.

1.1 Camera model

The pin-hole model is the simplest model for a camera. An image of an infinitesimally small point light source taken with a pinhole camera contains a single, infinitesimally narrow peak at the corresponding pixel location. We can model its frequency response as a Dirac function.

In a real camera several non idealities contribute to a significant deviation from the pin-hole model. The linear distortion introduced by the optics is represented by the point spread function (PSF). This is the impulse response

of the imaging system, i.e. the image obtained when a point light source of infinitesimal size is placed in front of the system. Even when the system is perfectly focused, the image is not a point of infinitesimal size, but rather a disk of non negligible diameter. This measure describes the quality of the optical system. For example, lenses that are not ideal or are not precisely placed, result in an increase of the size of the point spread function. However, even in the ideal case, the point spread function has a non-negligible size. For an ideal lens with circular aperture, the point spread function is also called the Airy disk [7]. Its size is determined by the diffraction of the system, which is proportional to the wavelength of the light source and the aperture value (or f-number). Note that higher f-numbers correspond to a smaller aperture area, or less incident light. A large f-number corresponds to a large Airy disk and a strong low-pass effect (and at the same time a large depth of field). Conversely, a small f-number corresponds to a smaller Airy disk and sharper images.

Similarly to the point spread function, an additional low-pass effect is introduced by the sensor. In fact, it is not possible to measure light intensity on a sampling point of infinitesimal size. Instead, a sensor integrates the amount of photons hitting the pixel surface. Such an integration (along space and time coordinates) corresponds to a low-pass effect that is proportional to the size of the integration surface. In the continuing quest for higher resolution, pixel sizes are reduced, and therefore the low-pass filtering effect is decreased. However, to increase light sensitivity, sensor manufacturers increase the fill factor, i.e. the active part of the pixels. Unfortunately, increasing the fill factor reduces the bandwidth of the system and limits the advantage of applying super-resolution algorithms.

Frequency responses $|H(\omega)|$ of some imaging systems are compared in Figure 1.2. The frequency scale is normalized with respect to the sampling frequency. The frequency responses are computed analytically, taking only the diffraction of the (ideal) optics and the spatial integration of the sensor into account (assuming a fill factor of 100%). For the sensor, we consider the case of a 10.1 Mpixel $4/3''$ sensor[1], typical of a high quality camera, and a 3.2 Mpixel, $1/2.5''$ sensor, which can be found in a handheld device, such as a mobile phone. In the case of the high resolution sensor, we consider two (circular) lens aperture values, namely $f/5.6$ and $f/2.8$, and for the low resolution sensor we use a lens aperture of $f/2.8$. Aliasing is necessary to any algorithm for super-resolution. Therefore, the frequency response has to be non-negligible for normalized frequency values larger than 0.5 cycles/pixel. This occurs for the case of the high resolution sensor and larger aperture optics, while for the other cases the aperture area is small with respect to the sensor resolution, resulting in a large Airy disk, such that the amount of

[1]Note that such size designations in fractional inches do not represent actual sensor sizes. This notation dates back to the 50's and TV camera tubes, where the size gives the outer diameter of the long glass envelope of the tube. The sensor diagonal is typically approximately 2/3 of this distance.

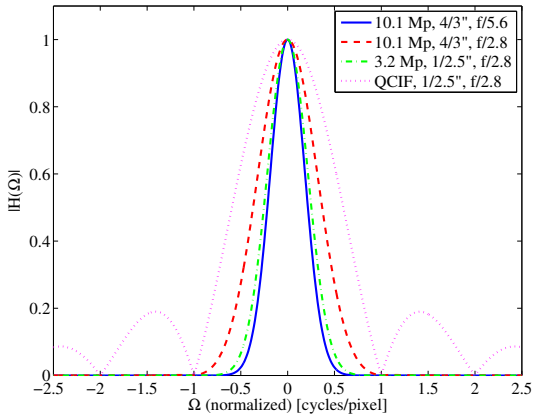


FIGURE 1.2
Comparison of frequency responses (PSF and sensor integration) for different imaging systems. Simulated frequency responses are shown for camera systems with a 10.1 Mpixel $4/3''$ sensor and $f/5.6$ and $f/2.8$ aperture of the optics, a 3.2 Mpixel $1/2.5''$ sensor with lens aperture $f/2.8$, and a QCIF video obtained by combining blocks of pixels from the 3.2 Mpixel $1/2.5''$ sensor.

aliasing is not significant. Moreover, we see that for the 10.1 Mpixel camera with aperture $f/2.8$, the response vanishes for normalized frequencies larger than 1 cycle/pixel.

An interesting case is the one where the low resolution sensor is used to acquire a QCIF video. In this case, the sensor resolution (3.2 Mpixel) is larger than the output resolution (144×176 pixels). Blocks of sensor pixels are combined in order to give an effect equivalent to a reduction of resolution (note that normally handheld devices do not apply any additional filter). This operation is equivalent to filtering the image with an averaging filter and then downsampling without any additional filtering. It results in a higher level of aliasing on the final images that compose the video. The equivalent response function is also shown in Figure 1.2. We remark that a significant amount of aliasing can be present in the range of frequencies between 0.5 and 1 (and even up to 2) cycles/pixel.

Finally, these modeled frequency responses were compared to the measured frequency responses of some practical systems (see Figure 1.3). First, we took a Leica DC250 grayscale digital camera that is often used in microscopy. It has a 8.6×6.9mm (or equivalently, $2/3''$) sensor, producing images of 1280×1024 pixels. We combined this camera with a Nikon 85mm lens using a C mount to Nikon adapter (no f-number available for this experiment). A considerable amount of aliasing is obtained in this setup, up to normalized frequencies of 1.2 cycles/pixel. As a second test camera, we used a Sigma SD10 digital camera with a Foveon X3 sensor (20.7×13.8mm, or equivalently about $4/3''$).

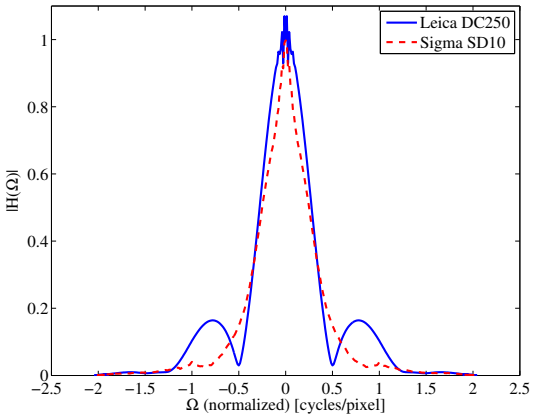


FIGURE 1.3
Measured frequency responses for a Leica DC250 camera with a Nikon 85mm optical system and a Sigma SD10 camera with a Sigma 18-50 mm lens.

This sensor measures the red, green and blue channel at each pixel position, taking away the need for a demosaicing color interpolation. We used a Sigma $18-50$ mm lens at a focal length of 35 mm with this camera with aperture $f/10$. The images captured are 2268×1512 pixels. The Sigma camera has a lower (relative) cutoff frequency, but still shows a non-negligible frequency response up to 0.8 cycles/pixel.

1.2 What is Resolution?

Before we dive into super-resolution algorithms to 'increase' the resolution, let us define what we understand by the term 'resolution'. There is definitely more to resolution than a simple pixel count, which is often simplistically used to indicate camera 'resolution'. Applying a low-pass filter to an image does much more to its resolution than merely increasing its number of pixels by repeating each pixel. Resolution relates to the ability to distinguish details in an image, in other words, to the resolving power.

In optics, the term 'optical resolution' is used as a measure of the ability of a camera system, or a component of a camera system, to depict picture detail [7]. Assuming a diffraction-limited lens, two point light sources are said to be just resolved if the center of one Airy disk coincides with the first minimum of the other Airy disk. This is called Rayleigh's criterium. Actually, this criterium slightly underestimates the resolution, and a better condition is given by Sparrow. It says that two point light sources can be resolved until

their two Airy disks overlap such that the second derivative at the center of one of the Airy disks is zero: the dip between the two Airy disks has disappeared. Similar criteria can also be applied to non-ideal optical systems. The role of the Airy disk is then taken over by the point spread function.

In imaging, we talk about image resolution as a measure of the amount of detail that is visible in an image. The International Organization for Standardization (ISO) has developed a precise method to measure the resolution of a digital camera system [10]. The visual resolution can be measured as the highest frequency pattern of black and white lines where the individual black and white lines can still be visually distinguished in the image. It is expressed in line widths per picture height (LW/PH). The standard also describes a method to compute the spatial frequency response of a digital camera. It describes the variation between the maximum and minimum values that is visible as a function of the spatial frequency (the number of black and white lines per millimeter). It can be measured using an image of a slanted black and white edge, and is expressed in relative spatial frequencies (relative to the sampling frequency), line widths per picture height, or cycles per millimeter on the image sensor. Figure 1.4 shows the resolution chart used in the ISO standard. Examples of measured spatial frequency responses are shown in Figure 1.3.

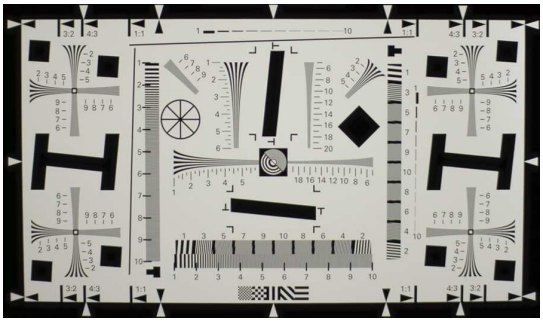


FIGURE 1.4
Resolution chart used to measure image resolution according to the ISO standard [10].

1.3 Super-Resolution as a Multichannel Sampling Problem

Let us now analyze super-resolution reconstruction mathematically, and formulate it as a multichannel sampling problem with unknown offsets. In order to keep the equations and analysis as simple as possible, we will present most

of the material for 1D signals, and only use 2D notations where needed. The extension to 2D signals is straightforward. For simplicity, we will also assume a pinhole camera model, except when explicitly specified.

Consider a (continuous-time) input signal $f(x)$ in an L-dimensional Hilbert space. We can write $f(x)$ as a linear combination of the Hilbert space basis functions:

$$f(x) = \sum_{l=0}^{L-1} \alpha_l \varphi_l(x), \tag{1.1}$$

with α_l the expansion coefficient corresponding to the l-th basis function $\varphi_l(x)$. In many cases, the Hilbert space will be the space of truncated Fourier series, but it can also be applied to other spaces such as splines, wavelets, etc.

We now sample this signal with N sample sets y_n $(0 \leq n < N)$ at a rate K, where each set is taken with an arbitrary offset t_n:

$$y_n(k) = f\left(\frac{k+t_n}{K}\right) = \sum_{l=0}^{L-1} \alpha_l \varphi_l\left(\frac{k+t_n}{K}\right). \tag{1.2}$$

This results in N sets of K uniformly spaced samples with offsets $\mathbf{t} = (t_0, t_1, \ldots t_{N-1})$. We can combine the samples for each set in a sample vector $\mathbf{y}_n$, the expansion coefficients α_l into a coefficient vector $\boldsymbol{\alpha}$, and the sampled basis functions $\varphi_l(x)$ into a matrix $\boldsymbol{\Phi}_{t_n}$, where $\boldsymbol{\Phi}_{t_n}(k,l) = \varphi_l\left(\frac{k+t_n}{K}\right)$. Equation (1.2) can then be rewritten as

$$\mathbf{y}_n = \boldsymbol{\Phi}_{t_n} \boldsymbol{\alpha}. \tag{1.3}$$

Stacking the different sample vectors and basis function matrices vertically, we obtain

$$\mathbf{y} = \boldsymbol{\Phi}_{\mathbf{t}} \boldsymbol{\alpha}. \tag{1.4}$$

In super-resolution imaging we typically want to reconstruct the original signal $f(x)$ (or equivalently, its coefficients $\boldsymbol{\alpha}$) from the images y_n $(0 \leq n < N)$. There are NK equations (1.2) in the L unknown signal coefficients and the $N-1$ offsets (without loss of generality, we can set $t_0 = 0$). This is exactly the same configuration as in multichannel sampling with unknown offsets. A reconstruction method for multichannel sampling with known offsets was presented by Papoulis [12]. We will mainly concentrate here on an accurate estimation of the offset values $\mathbf{t}$, which is an essential first step in accurate super-resolution.

As can be seen from (1.2), for general basis functions these equations are linear in the signal coefficients, but non-linear in the offset values. It can be shown that if $NK > L + N - 1$, the solution to this set of equations is unique (except in some degenerate cases) [17]. If less samples are available (either by taking less sample sets or by using sets of lower resolution), the problem is ill-posed, and an additional regularization is typically needed. In this chapter, we will mainly consider cases for which $NK > L + N - 1$.

1.3.1 Fourier series

Let us now analyze the above setup for the specific case of a truncated Fourier series:

$$f(x) = \sum_{l=-M}^{M} \alpha_l \varphi_l(x), \tag{1.5}$$

where the index l is now numbered from $-M$ to M because of the usual numbering for Fourier series ($L = 2M + 1$). The samples from (1.2) now become

$$y_n(k) = f\left(\frac{k + t_n}{K}\right) = \sum_{l=-M}^{M} \alpha_l e^{j2\pi \frac{l(k+t_n)}{K}} = \sum_{l=-M}^{M} \alpha_l W^{lk} z_n^l, \tag{1.6}$$

with $W = e^{j2\pi/K}$ and $z_n = e^{j2\pi t_n/K}$. As before, we can rewrite a sample set in matrix notation as

$$\mathbf{y}_n = \mathbf{F}^* \mathbf{D}_{t_n} \boldsymbol{\alpha}, \tag{1.7}$$

with

$$\mathbf{F} = \begin{pmatrix} 1 & W^M & \cdots & W^{(K-1)M} \\ \vdots & \vdots & & \vdots \\ 1 & W & \cdots & W^{K-1} \\ 1 & 1 & \cdots & 1 \\ 1 & W^{-1} & \cdots & W^{-(K-1)} \\ \vdots & \vdots & & \vdots \\ 1 & W^{-M} & \cdots & W^{-(K-1)M} \end{pmatrix} \tag{1.8}$$

$$\mathbf{D}_{t_n} = \begin{pmatrix} z_n^{-M} & & & & & & 0 \\ & \ddots & & & & & \\ & & z_n^{-1} & & & & \\ & & & 1 & & & \\ & & & & z_n & & \\ & & & & & \ddots & \\ 0 & & & & & & z_n^M \end{pmatrix}. \tag{1.9}$$

(1.10)

Note that $\mathbf{F}$ is an $L \times K$ forward DFT matrix, and the notation $\mathbf{F}^*$ is used to indicate the Hermitian transpose of $\mathbf{F}$, indicating the inverse DFT matrix. Due to the undersampling ($K < L$), some of the rows in $\mathbf{F}$ are repeated. The matrix $\mathbf{D}_{t_n}$ is an $L \times L$ diagonal matrix with its diagonal elements depending on the offset t_n. Just like for the general case, we can combine the different

sample sets into one vector, resulting in

$$\mathbf{y} = \begin{pmatrix} \mathbf{y}_0 \\ \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_{N-1} \end{pmatrix} = \begin{pmatrix} \mathbf{F}^* \\ \mathbf{F}^*\mathbf{D}_{t_1} \\ \vdots \\ \mathbf{F}^*\mathbf{D}_{t_{N-1}} \end{pmatrix} \boldsymbol{\alpha}. \tag{1.11}$$

The Fourier transform $\mathbf{y}_n^F$ of a sample set $\mathbf{y}_n$ can be computed as

$$\mathbf{y}_n^F = \frac{1}{K}\mathbf{F}_K\mathbf{y}_n = \frac{1}{K}\mathbf{F}_K\mathbf{F}^*\mathbf{D}_{t_n}\boldsymbol{\alpha}, \tag{1.12}$$

with $\mathbf{F}_K$ a square $K \times K$ (non-aliased) DFT matrix. As we know from sampling theory, $\mathbf{y}_n^F$ is an aliased and phase shifted version of the original Fourier coefficient vector $\boldsymbol{\alpha}$. This can be seen if we take for example $L = 3K$:

$$\begin{aligned} \mathbf{y}_n^F &= \frac{1}{K}\mathbf{F}_K\mathbf{F}^*\mathbf{D}_{t_n}\boldsymbol{\alpha} = \frac{1}{K}\mathbf{F}_K \begin{pmatrix} \mathbf{F}_K^* & \mathbf{F}_K^* & \mathbf{F}_K^* \end{pmatrix} \mathbf{D}_{t_n}\boldsymbol{\alpha} \\ &= \begin{pmatrix} I & I & I \end{pmatrix} \mathbf{D}_{t_n}\boldsymbol{\alpha} = \sum_{i=-1}^{1} z_n^{iK} \mathbf{D'}_{t_n}\boldsymbol{\alpha}_i, \end{aligned} \tag{1.13}$$

where $\mathbf{D'}_{t_n}$ is the $K \times K$ central part of the $L \times L$ matrix $\mathbf{D}_{t_n}$, and $\boldsymbol{\alpha}_i$ is the i-th block of K coefficients from $\boldsymbol{\alpha}$. In general, when L is not a multiple of K, we can still do the same decomposition by adding zeros to the vector $\boldsymbol{\alpha}$ up to the next multiple of K.

Depending on the sampling frequency K, we can consider three different cases (see Figure 1.5). If $K > L$, the signal/image is sampled according to the Nyquist sampling theorem, and no aliasing is present. From a super-resolution point of view, this case is not interesting, as there is no aliasing from which to extract additional high frequency information. If $M < K < 2M$, part of the frequency spectrum is aliased, leaving also part of the spectrum free of aliasing. Solutions for such a case will be discussed in Section 1.5. In the next section, we will analyze the third case, $K < M$, where the entire frequency spectrum is aliased. In such a situation, the registration parameters can only be determined accurately by jointly estimating them from the full set of images (as opposed to the common pairwise registration in other cases).

1.4 Registration of Totally Aliased Signals

1.4.1 Variable Projection Method

As discussed in Section 1.3, the equations from (1.4) are non-linear in the registration parameters, and linear in the signal expansion coefficients. We

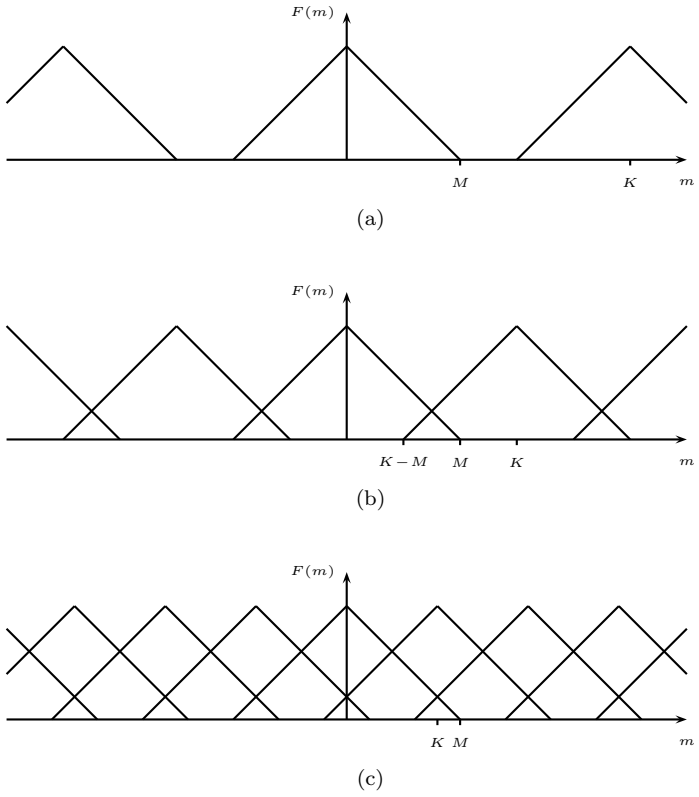


FIGURE 1.5
Three sampling situations can be distinguished: (a) Nyquist sampling ($K > 2M$), (b) Partially aliased signals ($M < K < 2M$), and (c) Totally aliased signals ($K < M$).

can write (1.4) as an l_2 minimization problem:

$$\min_{\boldsymbol{\alpha},\mathbf{t}} \|\mathbf{y} - \boldsymbol{\Phi}_{\mathbf{t}}\boldsymbol{\alpha}\|_2^2, \tag{1.14}$$

which is exactly the template problem in nonlinear least squares [6]. It can be solved using a variable projection method. For the correct values $\mathbf{t}$, the sample vector $\mathbf{y}$ is a linear combination of the sampled basis functions, represented by the columns of $\boldsymbol{\Phi}_{\mathbf{t}}$. In other words, $\mathbf{y}$ is in the subspace spanned by $\boldsymbol{\Phi}_{\mathbf{t}}$, so we can estimate $\mathbf{t}$ by minimizing the difference between $\mathbf{y}$ and its projection onto the estimated subspace:

$$\min_{\mathbf{t}} \|\mathbf{y} - \boldsymbol{\Phi}_{\mathbf{t}}(\boldsymbol{\Phi}_{\mathbf{t}}^*\boldsymbol{\Phi}_{\mathbf{t}})^{-1}\boldsymbol{\Phi}_{\mathbf{t}}^*\mathbf{y}\|_2^2. \tag{1.15}$$

This method for solving a nonlinear least squares problem is called the variable projection method [6]: the sample vector is actually projected on a variable subspace that depends on the minimization parameters.

Note that for the Fourier basis, this space can be split in a number of orthogonal subspaces corresponding to the different Fourier vectors. Aliased frequencies appear in the same subspace. The minimization can therefore be applied independently on each of those subspaces (where each subspace will have periodically repeating minima), and these can then be combined to obtain the joint minimum. Moreover, the subspaces containing as many aliased components as their dimensionality can be skipped, as any set of offsets will work here. The minimization over independent subspaces is illustrated in Figure 1.6.

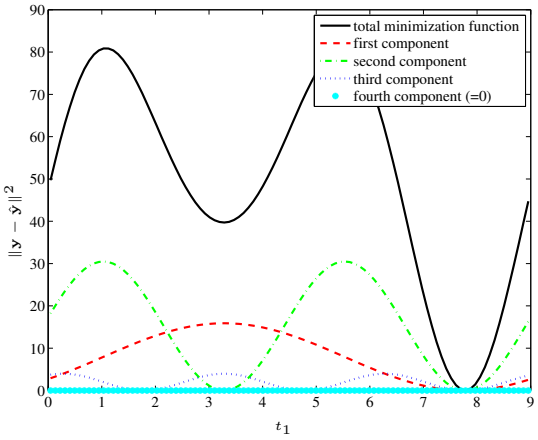


FIGURE 1.6
Example of the decomposition of the objective function into its different components belonging to orthogonal subspaces.

This registration method can be generalized to any type of motion model, as it only requires the basis functions $\varphi_l(t)$ to be sampled according to the model. Of course, more complex motion models will increase the dimensionality of the optimization problem.

1.4.2 Frequency Analysis Method

Let us now consider again an image in Fourier space. The equation from (1.13) can be generalized for arbitrary L to

$$\mathbf{y}_n^F = \mathbf{D}'_{t_n} \sum_{i=\lceil -(S-1)/2 \rceil}^{\lceil (S-1)/2 \rceil} z_n^{iK} \boldsymbol{\alpha}_i, \tag{1.16}$$

where $S = \lceil L/K \rceil$, or in other words, L was increased to the next multiple of K, and $\boldsymbol{\alpha}$ was split up accordingly into S parts $\boldsymbol{\alpha}_i$ of length K. If we multiply both sides of the above equation by $\mathbf{D'}_{t_n}^{-1}$, we obtain modified sample vectors

$$\mathbf{D'}_{t_n}^{-1}\mathbf{y}_n^F = \sum_{i=\lceil -(S-1)/2 \rceil}^{\lceil (S-1)/2 \rceil} z_n^{iK}\boldsymbol{\alpha}_i. \tag{1.17}$$

From this equation, it is clear that each modified sample vector $\mathbf{D'}_{t_n}^{-1}\mathbf{y}_n^F$ is part of the same S-dimensional subspace spanned by the spectrum vectors $\boldsymbol{\alpha}_i$. If we therefore take $N > S$ sample vectors, the matrix

$$\begin{pmatrix} \mathbf{y}_0^F & \mathbf{D'}_{t_1}^{-1}\mathbf{y}_1^F & \cdots & \mathbf{D'}_{t_{N-1}}^{-1}\mathbf{y}_{N-1}^F \end{pmatrix} \tag{1.18}$$

should be rank-deficient. These modified sample vectors depend on the offset values, and therefore we can estimate the registration parameters by searching the parameters that minimize the rank of the matrix in (1.18), or equivalently, minimize its $S+1$-th singular value σ_{S+1}:

$$\min_{\mathbf{t}} \sigma_{S+1} \begin{pmatrix} \mathbf{y}_0^F & \mathbf{D'}_{t_1}^{-1}\mathbf{y}_1^F & \cdots & \mathbf{D'}_{t_{N-1}}^{-1}\mathbf{y}_{N-1}^F \end{pmatrix} \tag{1.19}$$

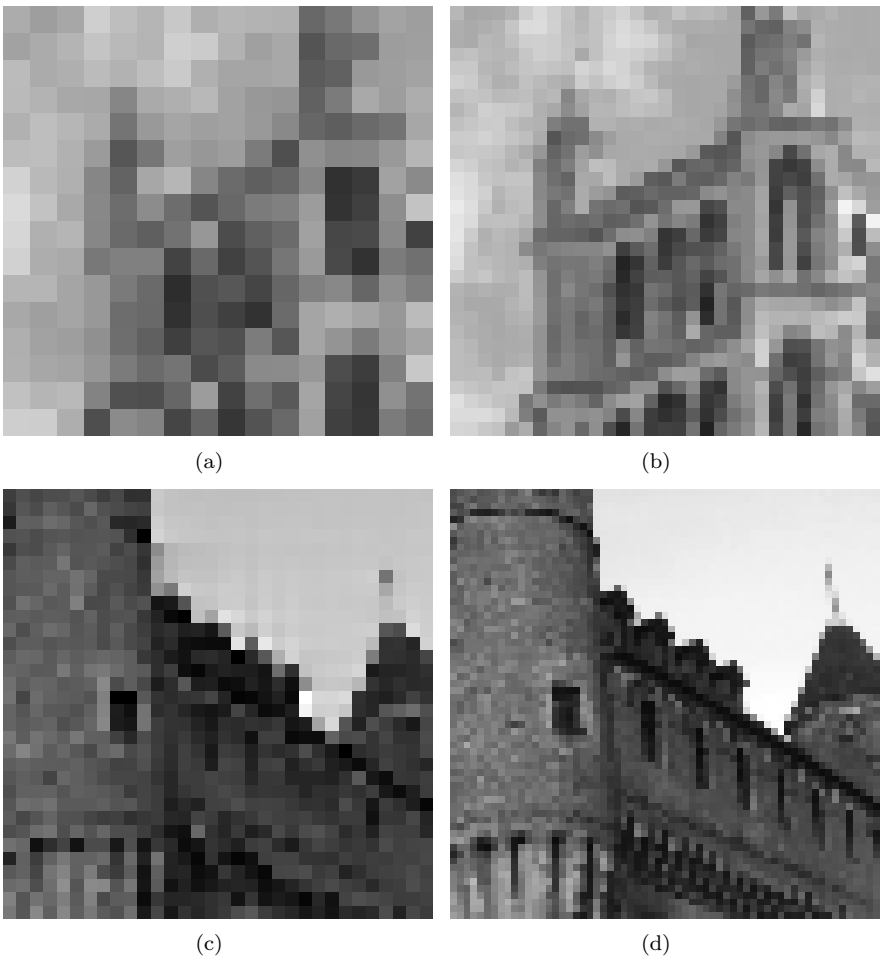
1.4.3 Results

The above algorithms were tested in numerical simulations. From an original image, we created a set of 5 low resolution images at half the resolution with relative (periodic) shifts (such that the entire spectrum is aliased). Two such input images can be seen in Figures 1.7(a) and 1.7(c). Both of the above techniques give correct estimation of the motion parameters, and allow perfect reconstruction of the original high resolution images 1.7(b) and 1.7(d).

The minimization required in the two described algorithms has a high computational complexity. In both cases, an $N-1$-dimensional function has to be minimized (or $2(N-1)$ for images with horizontal and vertical shifts), which has a large number of local minima. Examples of such minimization functions are shown in Figure 1.8. For some options on search algorithms to find the optimum and a more detailed complexity analysis, we refer the reader to [17].

1.5 Registration of Partially Aliased Signals

The algorithms presented in the previous section are generally applicable to super-resolution from aliased images. However, as discussed above, they also have a high computational complexity. In this section we present two methods with lower complexity in case only part of the frequency spectrum is aliased.



(a) (b)

(c) (d)

FIGURE 1.7
Simulation results of the algorithms for totally aliased images (noiseless). (a) One of the five 16×16 images used as input. (b) Reconstructed 31×31 image using the algorithm from Section 1.4.1. (c) One of the five 32×32 images used as input. (d) Reconstructed 63×63 image using the algorithm from Section 1.4.2.

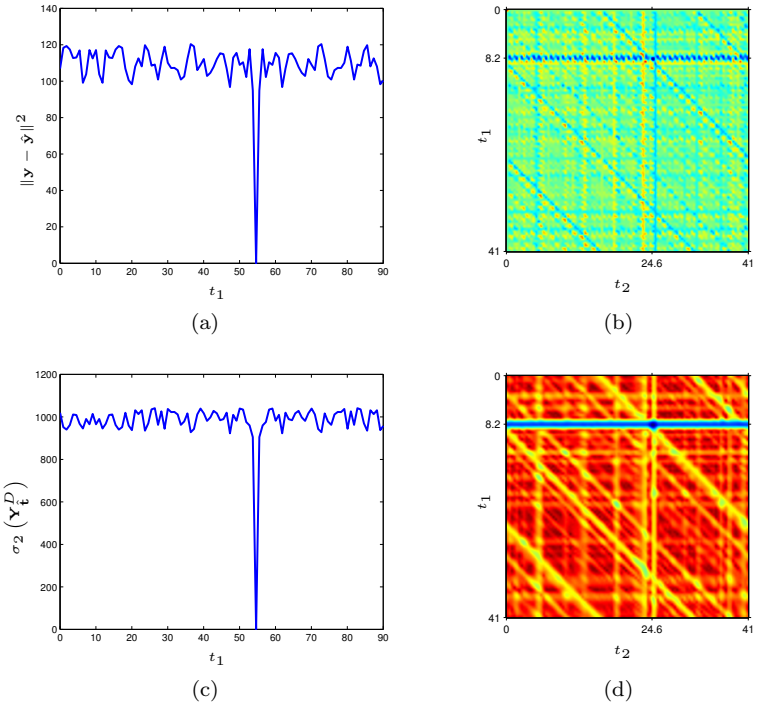


FIGURE 1.8
Examples of the objective functions in (1.15) and (1.19). (a) Two sets of 91 samples, with 81 unknown coefficients used in (1.15). The exact offset is $t_1 = 54.6$. Next to the global minimum, they also contain many local minima. (b) Three sets of 41 samples, with 81 unknown coefficients used in (1.15). The exact offsets are $t_1 = 8.2$ and $t_2 = 24.6$. Small values are represented by dark pixels. (c) Two sets of 91 samples, with 81 unknown coefficients used in (1.19). The exact offset is $t_1 = 54.6$. (d) Three sets of 41 samples, with 81 unknown coefficients used in (1.19). The exact offsets are $t_1 = 8.2$ and $t_2 = 24.6$. Small values are represented by dark pixels.

1.5.1 Super-resolution using frequency domain registration

If we assume $M < K < 2M$ (or equivalently $L/2 < K < L$), the signal is aliased, but not over the entire spectrum (see Figure 1.5). In such a case, we can use the aliasing-free part of the spectrum to estimate the registration parameters. Using these registration parameters, the aliased part can then be disambiguated and we can reconstruct a higher resolution signal.

1.5.1.1 Image registration

As $N > L/2$, part of the spectrum is free of aliasing. In (1.13) we can see that for certain frequencies,

$$\mathbf{y}_n^F(l) = z_n^l \boldsymbol{\alpha}(l), \tag{1.20}$$

or in other words, those frequency coefficients are phase shifted versions of the same coefficients for other images:

$$\frac{\mathbf{y}_n^F(l)}{\mathbf{y}_{n'}^F(l)} = \frac{z_n^l \boldsymbol{\alpha}(l)}{z_{n'}^l \boldsymbol{\alpha}(l)} = \frac{z_n^l}{z_{n'}^l}. \tag{1.21}$$

We can therefore estimate the registration parameters from all frequencies $l < K - L/2$. This can be done robustly by fitting a plane through the phase differences for each of those frequencies. Note that phase wrapping needs to be taken into account in such an approach. Such a registration method is equivalent to applying a low-pass filter to the images (removing all aliased frequencies) prior to registration.

The above approach can be used to estimate horizontal and vertical motion in a plane parallel to the image plane. We will now extend the frequency domain registration to rotations in the same image plane. In order to do this, we need to use two-dimensional notations:

$$f_1(\boldsymbol{x}) = f_0(\mathbf{R}(\boldsymbol{x} + \boldsymbol{x}_1)), \tag{1.22}$$

$$\text{with } \boldsymbol{x} = \begin{pmatrix} x_h \\ x_v \end{pmatrix}, \boldsymbol{x}_1 = \begin{pmatrix} x_{1,h} \\ x_{1,v} \end{pmatrix}, \mathbf{R} = \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 \\ \sin\theta_1 & \cos\theta_1 \end{pmatrix}.$$

This can be expressed in Fourier domain as

$$\begin{aligned} f_1^F(\boldsymbol{u}) &= \iint_{\boldsymbol{x}} f_1(\boldsymbol{x}) e^{-j2\pi \boldsymbol{u}^T \boldsymbol{x}} d\boldsymbol{x} \\ &= \iint_{\boldsymbol{x}} f_0(\mathbf{R}(\boldsymbol{x} + \boldsymbol{x}_1)) e^{-j2\pi \boldsymbol{u}^T \boldsymbol{x}} d\boldsymbol{x} \\ &= e^{j2\pi \boldsymbol{u}^T \boldsymbol{x}_1} \iint_{\boldsymbol{x}'} f_0(\mathbf{R}\boldsymbol{x}') e^{-j2\pi \boldsymbol{u}^T \boldsymbol{x}'} d\boldsymbol{x}', \end{aligned} \tag{1.23}$$

with $f_1^F(\boldsymbol{u})$ the two-dimensional Fourier transform of $f_1(\boldsymbol{x})$ and the coordinate transformation $\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{x}_1$.

The rotation can be estimated independently before the shift estimation,

as the amplitude of the Fourier transforms does not depend on the shift values (for the aliasing-free part of the spectrum):

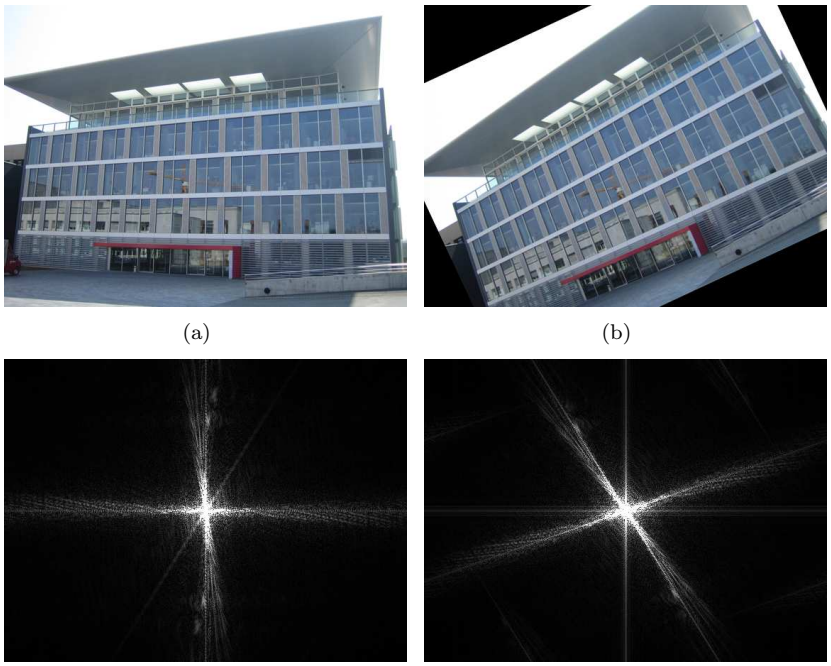
$$\begin{aligned} |f_1^F(\boldsymbol{u})| &= \left| e^{j2\pi \boldsymbol{u}^T \boldsymbol{x}_1} \iint_{\boldsymbol{x}'} f_0(\mathbf{R}\boldsymbol{x}') e^{-j2\pi \boldsymbol{u}^T \boldsymbol{x}'} d\boldsymbol{x}' \right| \\ &= \left| \iint_{\boldsymbol{x}'} f_0(\mathbf{R}\boldsymbol{x}') e^{-j2\pi \boldsymbol{u}^T \boldsymbol{x}'} d\boldsymbol{x}' \right| \\ &= \left| \iint_{\boldsymbol{x}''} f_0(\boldsymbol{x}'') e^{-j2\pi \boldsymbol{u}^T (\mathbf{R}^T \boldsymbol{x}'')} d\boldsymbol{x}'' \right| \\ &= \left| \iint_{\boldsymbol{x}''} f_0(\boldsymbol{x}'') e^{-j2\pi (\mathbf{R}\boldsymbol{u})^T \boldsymbol{x}''} d\boldsymbol{x}'' \right| \\ &= \left| f_0^F(\mathbf{R}\boldsymbol{u}) \right|, \end{aligned} \tag{1.24}$$

using the transformation $\boldsymbol{x}'' = \mathbf{R}\boldsymbol{x}'$. We can see that $|f_1^F(\boldsymbol{u})|$ is a rotated version of $|f_0^F(\boldsymbol{u})|$ over the same angle θ_1 as the spatial domain rotation (see also Figure 1.9). $|f_0^F(\boldsymbol{u})|$ and $|f_1^F(\boldsymbol{u})|$ do not depend on the shift values $\boldsymbol{x}_1$, because the spatial domain shifts only affect the phase values of the Fourier transforms. Therefore, we can first estimate the rotation angle θ_1 from the amplitudes of the Fourier transforms $|f_0^F(\boldsymbol{u})|$ and $|f_1^F(\boldsymbol{u})|$. After compensation for the rotation, the shift $\boldsymbol{x}_1$ can be computed from the phase difference between $f_0^F(\boldsymbol{u})$ and $f_1^F(\boldsymbol{u})$.

One option to estimate the rotation angle between two images is to compute the spectral differences of the aliasing-free frequencies for the reference image $f_0^F(\boldsymbol{u})$ with various rotations $f_1^F(\mathbf{R}\boldsymbol{u})$ of the image to register. However, this is a computationally very intensive method, as we need to compute rotations of the image spectra over a large number of rotation angles to find the optimal value. Instead, we will project the (aliasing-free) frequency content of the image onto a circular line and estimate the rotation angle by estimating the shift between two such one-dimensional functions. This is equivalent to transforming the image into polar coordinates and projecting this transform onto the axis associated to the angular coordinate.

1.5.1.2 Image reconstruction

After image registration, we reconstruct a high resolution image from the set of images using a non-uniform interpolation method implemented in Matlab [11]. Assuming the PSF is very narrow, and can be approximated by a Dirac, we compute the precise locations of all pixel coordinates on the high resolution grid. Next, we perform a Delaunay triangulation using the Quickhull algorithm [1]. The high resolution pixel values are then non-uniformly interpolated using bicubic interpolation. Such a reconstruction method provides good precision, with very low computational complexity. For more advanced reconstruction methods, we refer the reader to Chapters **??**.



(a) (b)

(c) (d)

FIGURE 1.9
The amplitude of the Fourier transform of an image is rotated over the same angle ($\theta_1 = 25°$) as the spatial domain image. (a) Original image. (b) Rotated image. (c) Fourier transform amplitude of the original image. (d) Fourier transform amplitude of the rotated image.

1.5.1.3 Results

Some results using the above algorithm are presented in Figures 1.10 and 1.11. Figure 1.10 shows a high resolution image reconstructed from 4 grayscale input images obtained using the Leica digital camera measured in Section 1.1. From the detail images, it is clear that more details can be observed in the reconstructed image than in any of the input images.

In a second experiment, we reconstructed a high resolution image from 4 color images taken with the Sigma camera from Section 1.1. Results for a patch of the image can be seen in Figure 1.11. Again, we can see that the aliasing has been accurately removed in the horizontal grids of the building. At the same time, a small mismatch can be seen on the branches of the tree in front. This is due to errors in the motion model: the planar motion parameters found for the building do not apply for the branches of the tree that move in the wind.

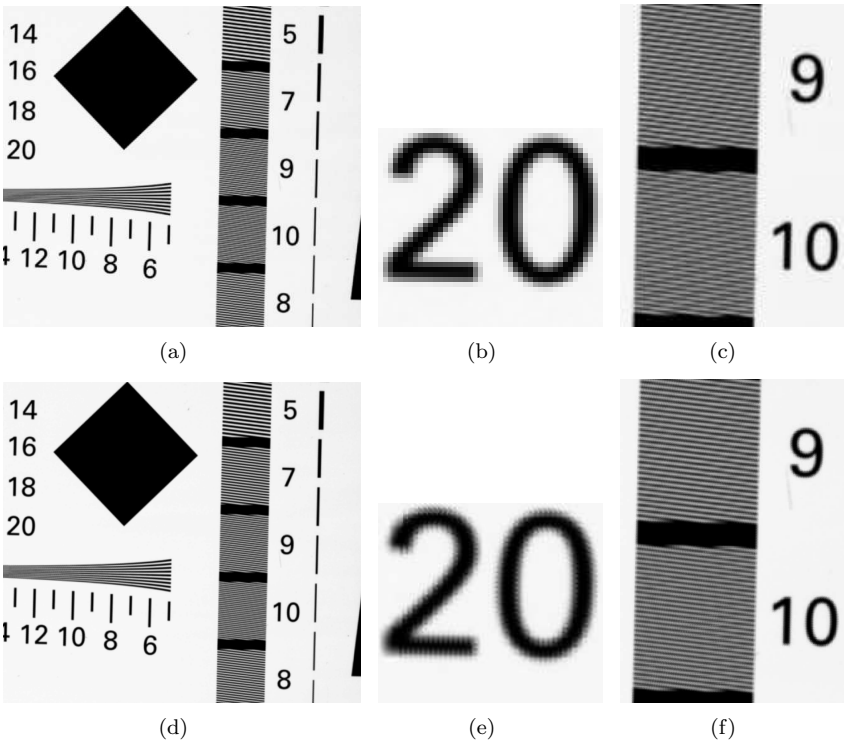


FIGURE 1.10
Results using the frequency domain registration algorithm for partially aliased images on a set of still images taken with a Leica DC250 camera. (a) Part of one of the input images, with a detail showing the aliasing in (b) and (c). (d) High resolution image reconstructed from part of four Leica DC250 input images. Details are shown in (e) and (f) to display the differences better.

(a) (b)

(c) (d)

FIGURE 1.11
Results of the frequency domain registration algorithm for partially aliased images on the Sigma SD10 images of the outdoor scene. (a) (Part of) one of the 4 input images, with a detail in (b). (c) High resolution output image, with a detail of the central part in (d) to show the differences better.

1.5.2 Super-resolution from low quality videos

In this section, we explore the possibility of applying super-resolution techniques to low resolution videos, such as those acquired with handheld devices. These devices typically acquire videos with low resolution (such as CIF, i.e. 288x352, or QCIF, i.e. 144x176) and the videos are coded at relatively low bit rates (such as 128 Kb/s). Despite the low quality of these videos, we will show that super-resolution algorithms can be applied, under certain hypotheses, to increase resolution and obtain additionally a significant reduction of coding artifacts.

As discussed in Section 1.1, the limitations of the acquisition system for the considered application are such that only the case of partial aliasing can be applied. Moreover, the type of motion present in a video can rarely be accurately modeled by a simple 2D translation or even a 2D roto-translation. This forces us to consider more general motion models and prevents the application of algorithms based on the Fourier transform alone.

1.5.2.1 Motion model

In the case of a video sequence, the movement of the camera can rarely be approximated by a simple 2D (roto)-translation. For this reason, we consider here some more general motion models. The parameters of these models have to be computed with high precision during the registration phase, such that the displacements in the image plane are much smaller than a pixel. In order to do this robustly, we restrict the choice of the motion model to those with few parameters. We used the 3D rotational and the planar model under the hypothesis of perspective projection. The 3D rotational model describes the motion in the scene as a 3D rotation around the camera. It is a good approximation when the objects present in the scene are distant from the camera or the translational component is negligible. The planar model approximates the scene with a planar surface in 3D space and the camera undergoes an arbitrary (3D) roto-translation. In both cases, the relation between a point at position p in the reference image and p' in one of the non reference images is given by [5]

$$\mathbf{p}' = \mathbf{K}\mathbf{H}\mathbf{K}^{-1}\mathbf{p}, \tag{1.25}$$

where $\mathbf{p} = [x\ y\ f]^T$ and $\mathbf{p}' = [x'\ y'\ f]^T$ are the homogeneous coordinates of the two points and f is the focal length. The matrix $\mathbf{K}$ is the camera calibration matrix and has the structure,

$$\mathbf{K} = \begin{bmatrix} \alpha & 0 & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{1.26}$$

where α and β are the magnification factors (i.e. the number of pixels per meter along the horizontal and vertical direction of the sensor) and $[u_0\ v_0]$ is the position of the principal point, i.e. the position of the point where the

optical axis intersects the sensor. The matrix $\mathbf{H}$ is a rotation matrix in the case of a 3D rotation centered at the focal point. In this case, no assumption is needed about the structure of the scene, since there is no parallax. In the case of a planar scene and an arbitrary roto-translation, $\mathbf{H}$ takes the structure of a homography matrix and has 8 degrees of freedom, since it is defined up to a scaling factor.

In the following, we assume that the parameters f, α, β, u_0, and v_0 are known. Their values can be determined precisely using a calibration procedure or they can be estimated based on the type of camera.

1.5.2.2 Image registration

Aliasing typically perturbs registration and its influence should be reduced for accurate results. A first way to achieve this consists in applying a low-pass filter (or only considering low frequencies) as done in Section 1.5.1. This reduces the spectral components associated to large aliasing amplitude. Another method is to operate in the spatial domain using robust estimators. The main idea is that the effect of aliasing is typically most visible along image edges, which are well localized in space. These regions give large registration errors and perturb algorithms that minimize the mean squared error (MSE). The problem is illustrated in Figure 1.12 for a simple one-dimensional case. The original continuous-time signal is represented in Figure 1.12(a). It consists of a low frequency component (a sinusoid) and a step function, which is non-bandlimited and represents an image edge. Two sets of samples are taken from the continuous time signal with a relative shift of $\delta = 0.083$ and are used to obtain an approximation of the input signal. The shift between the two sets is unknown and should be determined by a registration algorithm. A way to estimate the shift is to interpolate the sets of samples using a low-pass filter, as in Figure 1.12(b) and Figure 1.12(c), and determine the shift that minimizes the signal difference (MSE) between the two. However, this solution is not necessarily the correct one. For our example, such a minimization results in a shift $\delta = 0.25$ (while the correct shift was $\delta = 0.083$). Figure 1.12(e) shows the difference (error) between the interpolated signals in Figure 1.12(b) and Figure 1.12(c) for the estimated shift $\delta = 0.25$, while the difference for the correct shift is given in Figure 1.12(d). The reason for this behavior is the presence of the discontinuity, which implies large registration errors. The minimum MSE method tends to minimize the error at all points, irrespectively of their amplitude. Instead, we see that the correct shift presents large errors in the region of the discontinuity, but much smaller errors for the remaining points. This type of problem has been previously addressed in statistics and in image registration leading to the development of robust algorithms [8, 14].

As shown in Figure 1.1, we call the reference image I_0 and the other images I_n, $n = \pm 1, \pm 2, \ldots, \pm(N-1)/2$, assuming that the size of the circular buffer is N. A way to implement a robust estimator is to use an M-estimator and

minimize the quantity

$$J_n = \sum_{\mathbf{p}} e(I_0(\mathbf{p}) - w(I_n(\mathbf{p}), \mathbf{h}_n)), \tag{1.27}$$

for each non reference image. The function $w(I_n(\mathbf{p}), \mathbf{h}_n)$ computes a warped version of $I_n(\mathbf{p})$, i.e. an image for which the movement of the camera has been compensated. The vector $\mathbf{h}_n$ is a parametric representation of the matrix $\mathbf{H}$ (for example, the three Euler angles in the case of a 3D rotation). The function e is used to measure how close the reference and the motion compensated image are. For example, in the case of the MSE, $e(x) = x^2$, i.e. the error on each pixel contributes to the total error with the square of its value. The optimal choice of the function e follows the Maximum Likelihood principle (ML) which determines e according to the Probability Distribution Function (PDF) of the residual error [8]. The MSE measure is optimal in the case of a Gaussian distribution of the residual error. Instead, when outliers are present, such as in the case of some forms of aliasing, large errors occur proportionally more often and a slowly decaying PDF is a more appropriate model. In our implementation, the cost functions J_n are minimized using a multiresolution Gauss-Newton descent method, similar to that described by Sawhney and Ayer [15].

Some registration error statistics are given in Table 1.1 for estimators designed for Cauchy and Gaussian distributions of the residual error. Additionally, a pre-filter can be applied to reduce aliasing at high frequency components. The values are obtained by running 100 simulations with random motion parameters (for the 3D rotational model) corresponding to a pixel displacement in the range $[-0.5,\ 0.5]$.

The image used for the simulation is shown in Figure 1.13(a). A part of the image is downsampled to obtain the image in Figure 1.13(b) which is used as a reference image for the 100 runs of the registration algorithm. One of the 100 non reference images is shown in Figure 1.13(c). Aliasing is clearly visible in the region of the regular pattern of the car radiator.

The results in Table 1.1 show that the registration algorithm achieves subpixel precision. Both the low-pass filter and the robust estimator reduce the average registration error with respect to the Gaussian estimator. However, the low-pass filter is more effective than the robust estimator. This can be explained by the small amount of aliasing present in the images and the use of a Cauchy distribution for the residual error. Even if this is a better model than the Gaussian, it does not necessarily match the actual error distribution.

1.5.2.3 Image reconstruction

When the input images are registered with respect to the reference I_0, it is possible to determine the super-resolution image I_0' corresponding to I_0. We follow an approach similar to Farsiu et al. [4]. The procedure consists in

Estimator type	Avg. abs. error (pixels)	Max abs. error (pixels)
Gaussian	0.0421	0.0756
Cauchy	0.0224	0.0488
Gaussian with prefilter	0.00556	0.0184
Cauchy with prefilter	0.00488	0.0127

TABLE 1.1
Registration errors for different estimators with and without prefilter. The estimation errors are computed on 100 simulations using a 3D rotational model with parameters corresponding to displacements in the range $[-0.5,\ 0.5]$ pixels. Both the prefilter and the use of robust estimators contribute to the reduction of the registration errors.

minimizing, with respect to the unknown image I'_0, the cost function

$$J_S = \sum_n \sum_{\mathbf{p}} e(I_n(\mathbf{p}) - d \circ g \circ w(I'_0(\mathbf{p}, \mathbf{h}_n)) + \alpha T(I'_0), \tag{1.28}$$

where I_n are the non reference images, and the function $d \circ g \circ w$ represents the composition of down-sampling, low-pass filtering and warping. These functions represent the transformations that one should apply to the super-resolution image I'_0 to obtain each of the non reference images. This is realized by warping I'_0 according to the parameters computed in the registration step, and then low-pass filtering the result to simulate the behavior of camera optics, sensor, and motion blur. The last step is downsampling to reduce the number of pixels to that of the image I_n. The function e measures the error of the result with respect to the image I_n. When the image I'_0 is correct, the error should be small. As in the case of registration, the function e should be chosen according to the distribution of the residual errors. Typical choices are the L_1, L_2, and L_p norms. The additional term $\alpha T(I'_0)$ in equation (1.28) is needed to impose regularity on the image I'_0, because in most cases, the input images are not sufficient to determine the solution unambiguously. The need for regularization depends on the type of motion and the low-pass filter g. For example, it is not possible to determine the components of I'_0 that correspond to zeros of g. The typical choice for T is the Total Variation (TV) measure:

$$T(I'_0) = \int \sqrt{\|\nabla I'_0\|^2 + \beta}, \tag{1.29}$$

where β is a regularization term that makes T differentiable (we used $\beta = 0.01$ for frames with values in the range $[0,\ 255]$). The value of the Total Variation is related to the gradient magnitude of the image intensity, which is a measure of the image sharpness. In this way, the second term of equation (1.28) limits the amount of high frequencies added by the algorithm. In a software implementation of the Total Variation, the gradient operator in equation (1.29) is replaced by a difference operator. In our experiments, we approximated the

derivatives with the average of the forward and backward differences along the x and the y coordinates. The constant α in (1.28) controls the trade-off between regularity and level of details in the output image.

The minimization of J_S is performed on the space of the possible images I'_0, which has a dimension equal to the number of pixels. To limit the complexity of the algorithm, the steepest-descent method is used. This consists in applying iterations

$$I_0'^{(i+1)} = I_0'^{(i)} + \mu \frac{dJ_S}{dI'_0}, \tag{1.30}$$

to an initial guess $I_0'^{(0)}$. The step size μ determines the speed of convergence.

1.5.2.4 Results on video sequences

The proposed algorithm has been applied to a set of videos acquired using handheld devices and coded at low bit rates (around 128 Kb/s). The size of the circular buffer was $N = 9$. The motion parameters were computed assuming a 3D rotation, which is a good approximation when the distance of the scene is much larger than the translation of the camera. The large coding errors were the main source of noise during the frame registration step. This strongly reduced the advantage of the low-pass filter and robust estimation to minimize the effects of aliasing. However, robust estimators reduced the influence of regions not following the motion model, for example due to objects moving in the scene or model mismatch (such as optical distortion or non negligible translation). The descent method was applied to a multiscale representation of the frames. We found that 3 resolution levels and a maximum of 50 iterations were sufficient to obtain sub-pixel precision of the registrations. The super-resolution images were computed using the iterations from (1.30) with a step size $\mu = 0.07$. The term α in equation (1.28) determines the influence of the regularization term. A too small value will result in terms corresponding to the zeros of the filter g appearing in the solution. On the other hand, if α is too large, the improvement given by the algorithm is reduced. We found that $\alpha = 5$ was a good trade-off for our setup.

Some frames of the processed sequences are shown in Figure 1.14. The left column shows one of the original frames and the right column contains the super-resolution results. For comparison, the second column gives the result of bicubic interpolation. The increased level of details and sharpness is clearly visible on the super-resolution frames. In addition, we notice that some coding artifacts, like blockiness, are reduced in the output images. The reason is the temporal filtering introduced by the super-resolution algorithm, which tends to remove uncorrelated errors in the input images.

It is interesting to notice that, despite the high compression rate of the input videos, enough aliasing is still present in the video to be able to apply super-resolution. The reason is that the motion compensation step of a conventional video coder uses a block-based translational model, while the motion of the sequence is better modeled by a 3D rotation. Video encoders generally

encode a subset of (intra) frames directly, and use motion compensation to predict the (inter) frames in between. If the motion model is accurate, the inter frames can be predicted correctly, and there is no need to encode any residual error. This would result in a single frame and a set of motion vectors, and it would be impossible to apply super-resolution. As the 3D rotational motion is not well modeled by the translational model, the encoder has to spend a large portion of its bit rate to represent the residual error, which includes errors due to both motion model mismatch and aliasing. Therefore, we can conclude that it is the inefficiency of (motion compensation in) current video coders that makes super-resolution possible.

1.6 Conclusions

We have presented a set of super-resolution algorithms, ranging from a more theoretical analysis of super-resolution as a multichannel sampling problem with unknown offsets to a practical algorithm for low-resolution videos captured with a mobile phone. A special emphasis was given to the registration part of such algorithms, as a precise sub-pixel registration is a necessary prerequisite for a good reconstruction of additional details. Through simulations and practical experiments, we have shown the good performance of our algorithms in the relevant use cases.

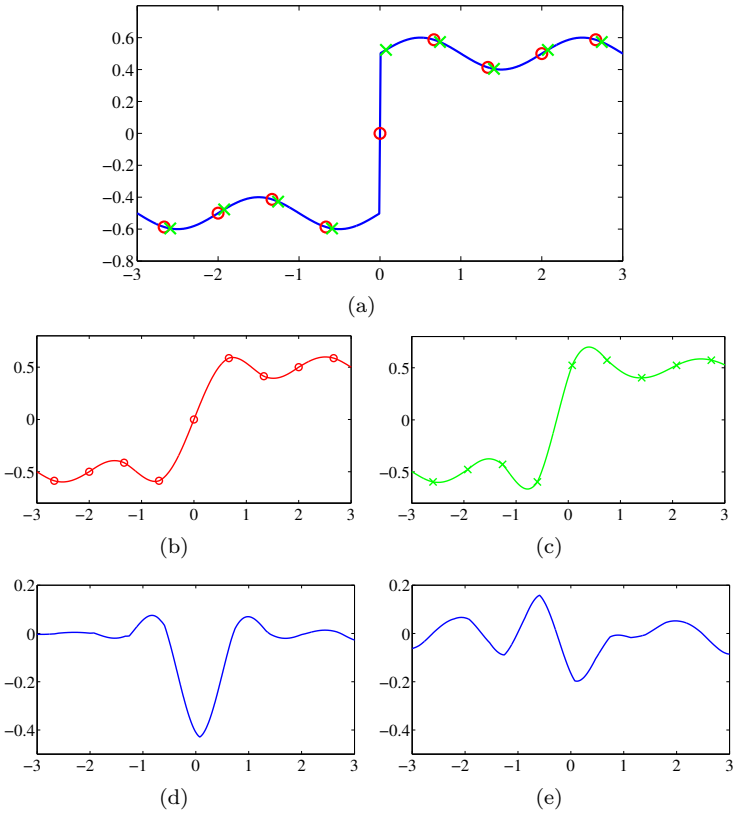
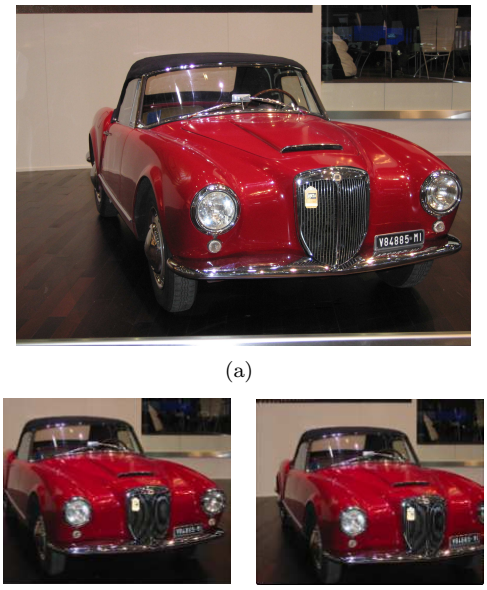


FIGURE 1.12
Example of signal registration for a 1D case. (a) A signal composed of a sinusoid and a step function is uniformly sampled twice with an unknown shift δ (in this example $\delta = 0.083$). The shift between the two sets of samples should be determined to reconstruct the original signal. (b) and (c) Low-pass interpolation of the two sets of samples. (d) Difference between the signals from (b) and (c) for the correct shift ($\delta = 0.083$). The error is large around the discontinuity (outlier) and small elsewhere. (e) Difference between the signals in (b) and (c) for the shift obtained using MSE minimization ($\delta = 0.25$). The error is minimized over the entire domain, but an incorrect shift value is obtained.



(a)

(b)

(c)

FIGURE 1.13
Images used to test the motion estimation algorithm. (a) Original high resolution (3072×2304) image. (b) Low resolution reference image (176×144). (c) One of the 100 low-resolution non reference images (176×144) registered with the reference image in (a).

(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

FIGURE 1.14
Results of super-resolution on low-rate encoded videos. The first column shows one video frame, the second column shows the result of bicubic interpolation, and the third column the result of the super-resolution algorithm applied to 9 consecutive video frames.

Bibliography

[1] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa. The Quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)*, 22(4):469–483, December 1996.

[2] S. Borman and R. Stevenson. Spatial resolution enhancement of low-resolution image sequences - a comprehensive review with directions for future research. Technical report, University of Notre Dame, 1998.

[3] EURASIP Journal on Applied Signal Processing, Special Issue on Super-Resolution, 2006.

[4] S. Farsiu, M. Elad, and P. Milanfar. Multiframe demosaicing and super-resolution of color images. *IEEE Transactions on Image Processing*, 15(1):141–159, January 2006.

[5] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, August 2002.

[6] G. Golub and V. Pereyra. Separable nonlinear least squares: the variable projection method and its applications. *Inverse Problems*, 19(2):R1–R26, 2003.

[7] E. Hecht. *Optics*. Pearson - Addison Wesley, 2002.

[8] P. J. Huber. *Robust Statistics*. John Wiley and Sons, 1981.

[9] IEEE Signal Processing Magazine, Special Issue on Super-Resolution, May 2003.

[10] International Organization for Standardization. ISO 12233:2000 - Photography - Electronic still picture cameras - Resolution measurements, 2000.

[11] Mathworks (The). Matlab function reference: griddata, 2009.

[12] A. Papoulis. Generalized sampling expansion. *IEEE Transactions on Circuits and Systems*, 24(11):652–654, November 1977.

[13] S. C. Park, M. K. Park, and M. G. Kang. Super-resolution image reconstruction: a technical overview. *IEEE Signal Processing Magazine*, 20(3):21–36, May 2003.

[14] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection*. Wiley, 1986, 2003.

[15] H. Sawhney and S. Ayer. Compact representations of videos through dominant and multiple motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:814–830, 1996.

[16] R. Y. Tsai and T. S. Huang. Multiframe image restoration and registration. In T. S. Huang, editor, *Advances in computer vision and image processing*, volume 1, pages 317–339. JAI Press, 1984.

[17] P. Vandewalle. *Super-Resolution from Unregistered Aliased Images*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, Switzerland, July 2006. No. 3591, [Reproducible] http://rr.epfl.ch/6.

[18] P. Vandewalle, L. Sbaiz, J. Vandewalle, and M. Vetterli. Super-Resolution from Unregistered and Totally Aliased Signals using Subspace Methods. *IEEE Transactions on Signal Processing*, 55(7, Part 2):3687–3703, 2007. [Reproducible] http://rr.epfl.ch/4.

[19] P. Vandewalle, S. Süsstrunk, and M. Vetterli. A Frequency Domain Approach to Registration of Aliased Images with Application to Super-Resolution. *EURASIP Journal on Applied Signal Processing, Special Issue on Super-Resolution Imaging*, 2006, 2006. Article ID 71459, 14 pages, [Reproducible] http://rr.epfl.ch/3.